An Efficient Attack on a Code-Based Signature Scheme

Aurélie Phesso, Jean-Pierre Tillich

An Efficient Attack on a Code-based Signature Scheme

Aurélie Phesso[2] Jean-Pierre Tillich[1]

[1] SECRET Project - INRIA Rocquencourt
Domaine de Voluceau, B.P. 105 78153 Le Chesnay Cedex - France
aurelie.phesso@gmail.com, jean-pierre.tillich@inria.fr
[2] Université Bordeaux
France.

Abstract. Baldi et al. have introduced in [BBC$^+$13] a very novel code based signature scheme. However we will prove here that some of the bits of the signatures are correlated in this scheme and this allows an attack that recovers enough of the underlying secret structure to forge new signatures. This cryptanalysis was performed on the parameters which were devised for 80 bits of security and broke them with $100,000$ signatures originating from the same secret key.

1 Introduction

It is a long standing open problem to build an efficient and secure signature scheme based on the hardness of decoding a linear code which could compete in all respects with DSA or RSA. Such schemes could indeed give a quantum resistant signature for replacing in practice the aforementioned signature schemes that are well known to be broken by quantum computers. The first answer to this question was given in [CFS01]. They adapted the Niederreiter scheme [Nie86] for this purpose. This requires a linear code for which there exists an efficient complete decoding algorithm. This means that if $\boldsymbol{H}$ is a $r \times n$ parity-check matrix of the code, there exists for any $\boldsymbol{s} \in \{0,1\}^r$ an efficient way to find a word $\boldsymbol{e}$ of smallest Hamming weight such that $\boldsymbol{H}\boldsymbol{e}^T = \boldsymbol{s}^T$. To sign a message $\boldsymbol{m}$, a hash function $\mathcal{H}$ is first applied to the message (say that the output of the hash function is a binary string of length r). Then the complete decoding algorithm of the code with parity-check matrix $\boldsymbol{H}$ is used to produce the signature of $\boldsymbol{m}$ which is a word $\boldsymbol{e}$ of smallest weight such that

$$\boldsymbol{H}\boldsymbol{e}^T = \mathcal{H}(\boldsymbol{m})^T.$$

The authors of [CFS01] noticed that very high rate Goppa codes are able to fulfill this task, and their scheme can indeed be considered as the first practical solution to the aforementioned problem. Moreover they gave a security proof of their scheme relying only on the assumption that two problems were hard, namely (i) decoding a generic linear code and (ii) distinguishing a Goppa code from a random linear code with the same parameters. However, afterwards it was realized that the parameters proposed in [CFS01] can be attacked by an unpublished attack of Bleichenbacher, which despite its exponential complexity gives an attack which is probably implementable in practice nowadays. Subsequently, it was shown in [Fin10] that there is a slight variation called Parallel-CFS which avoids the significant increase of parameters needed to thwart the Bleichenbacher attack on the original system. However, even this modified scheme shares the same nice features as the original scheme, that is very short signature sizes and reasonably fast software implementation for 80 bits of security [LS12] it has also some drawbacks, such as for instance:
(i) a lack of security proof in light of the distinguisher of high rate Goppa codes found in [FGO$^+$11] (see also [FGO$^+$13] for more details) which shows that the hypotheses used in [CFS01] to give a security proof of the signature scheme were not met,
(ii) and poor scaling of the parameters when security has to be increased. This comes from the following behavior. The [CFS01] scheme uses t-error correcting Goppa codes of length 2^m. The public key is of size $K = tm2^m$ whereas decoding attacks take about $\lambda = 2^{tm/2}$ operations whereas

obtaining signature needs about $t!t^2m^3$ operations. If we want to stick to a reasonable signature cost, this needs that we fix t to a small value (say smaller than 12). In this case the security parameter λ is basically only a polynomial function of the key size $K : \lambda \approx K^{t/2}$.

Other signature schemes based on codes were also given in the literature such as for instance the KKS scheme [KKS97,KKS05] or its variant [BMS11]. But they can be considered at best to be one-time signature schemes in light of the attack given in [COV07] and great care has to be taken to choose the parameters of this scheme as shown by [OT11] which broke all the parameters proposed in [KKS97,KKS05,BMS11].

Recently, there has been some revival in the CFS strategy [CFS01], by choosing other code families (or by replacing the Hamming metric by another metric). The new code families that were used are LDGM codes in [BBC$^+$13], i.e. codes with a Low Density Generator Matrix, LRPC codes in [GRSZ14], or (essentially) convolutional codes [GSJB14]. While there are still some doubts that there is a way to choose the parameters of the scheme [GSJB14] in order to avoid the attack [LT13] on the McEliece cryptosystem based on convolutional codes [LJ12], there was no clear indication that the two other schemes are insecure. In particular, the LRPC-based scheme comes with a security proof that obtaining a fairly large amount of message-signature pairs does not simplify the work of an attacker (and obtaining a feasable attack on the parameters proposed in [GRSZ14] is a completely open question). Both schemes are based on two very original ideas for decoding in the rank metric case for LRPC codes and decoding in the Hamming metric for the [BBC$^+$13] scheme.

The [BBC$^+$13] scheme builds upon the following idea. It is namely easy to find an error $\boldsymbol{e}$ of low weight which has some specific syndrome $\boldsymbol{s}$ (i.e. $\boldsymbol{s}^T = \boldsymbol{H}\boldsymbol{e}^T$) of low weight when the parity-check matrix $\boldsymbol{H}$ is systematic, i.e. it has the form $\boldsymbol{H} = \begin{pmatrix}\boldsymbol{P}\ \boldsymbol{I}\end{pmatrix}$, where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{P}$ is arbitrary. Here is enough to take $\boldsymbol{e} = \boldsymbol{0}||\boldsymbol{s}$ where $||$ stands for concatenation and $\boldsymbol{0}$ is the all-zero vector that has as many entries as there are columns in $\boldsymbol{P}$. Basically the authors of [BBC$^+$13] choose a hash function $\mathcal{H}$, such that the result of the hash function is a word of low weight for which the aforementioned decoding procedure works. Of course, an attacker can also perform the same task and it is the purpose of the [BBC$^+$13] scheme to hide the structure that allows this way of signing. This is obtained by taking LDGM codes whose low weight codewords will be used to hide the structure of the signature and by multiplying $\boldsymbol{H}$ by appropriate matrices.

However, contrarily to [GRSZ14] the LDGM code based scheme does not come with a security proof that message-signature pairs do not leak information. It is the purpose of this paper to show that indeed there is an efficient attack for breaking this scheme when the attacker has at her/his disposal enough signatures obtained from the same secret key. It is based on the fact that in this scheme some of the bits of the signature are correlated. These correlations can be used to recover an equivalent secret key which can be used to forge new signatures. This cryptanalysis was performed on the parameters which were devised for 80 bits of security and broke them with $100,000$ signatures originating from the same secret key in about one hour.

Notation: In the whole paper, the sum between bits is always performed as the sum over $\mathbb{F}_2$ (that is always modulo 2) and the sum between binary words $\boldsymbol{x} = (x_i)_i$ and $\boldsymbol{y} = (y_i)_i$ of the same length is performed componentwise $\boldsymbol{x}+\boldsymbol{y} = (x_i+y_i)_i$. We use bold letters for matrices and vectors, $\boldsymbol{A}$, $\boldsymbol{x}$ and so on and so forth. Vectors are understood as row vectors and we use the transpose notation to denote column vectors, for instance when $\boldsymbol{x}$ is a (row)-vector, $\boldsymbol{x}^T$ denotes this vector written in column form.

2 Description of the LDGM code based signature scheme proposed in [BBC$^+$13]

This scheme can be described as follows.

Private key.

- a full rank $k \times n$ binary matrix $\boldsymbol{G}$ with rows of some small and constant weight w_G which is a generator matrix of a binary LDGM code $\mathscr{C}$ of length n and dimension k. It is assumed that

the square $k \times k$ submatrix formed by the k first columns of $\boldsymbol{G}$ is invertible. In this case $\mathscr{C}$ admits an $(n-k) \times n$ parity-check matrix $\boldsymbol{H}$ of the form $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{P} & \boldsymbol{I} \end{pmatrix}$ where $\boldsymbol{I}$ is the identity matrix of size $(n-k) \times (n-k)$.

- an $n \times n$ matrix $\boldsymbol{S}$ that is sparse and non-singular of average row and column weight m_S.
- an invertible $(n-k) \times (n-k)$ transformation matrix $\boldsymbol{Q}$ of the form $\boldsymbol{Q} = \boldsymbol{R} + \boldsymbol{T}$ where $\boldsymbol{R}$ is of very low rank z (say 1 or 2) and $\boldsymbol{T}$ is sparse with row and column weight m_T. $\boldsymbol{R}$ can be written as $\boldsymbol{R} = \boldsymbol{a}^T \boldsymbol{b}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ are two $z \times (n-k)$ matrices.

Public key.

$$\boldsymbol{H}' = \boldsymbol{Q}^{-1} \boldsymbol{H} \boldsymbol{S}^{-1}.$$

Moreover this scheme also uses two fixed functions, a hash function $\mathcal{H}$ and a map $\mathcal{F}$ mapping any hash to a binary string of length $n-k$ and Hamming weight w.

Signature generation.

1. To sign a message $\boldsymbol{m}$, the signer computes $\boldsymbol{s} = \mathcal{F}(\mathcal{H}(\boldsymbol{m}))$ which is an element of $\{0,1\}^{n-k}$ of weight w. He checks whether $\boldsymbol{b}\boldsymbol{s}^T = 0$. If this is the case, he goes to the next step. If not, he appends a counter l to $\mathcal{H}(\boldsymbol{m})$ to obtain $\mathcal{H}(\boldsymbol{m})||l$ [3] and computes $\mathcal{F}$ applied to $\mathcal{H}(\boldsymbol{m})||l$ until getting a syndrome $\boldsymbol{s}$ of weight w that satisfies $\boldsymbol{b}\boldsymbol{s}^T = 0$ (for more details see [BBC+13, §3.2]). This requires $O(2^z)$ attempts on average.
2. The signer computes the private syndrome $\boldsymbol{s}'^T = \boldsymbol{Q}\boldsymbol{s}^T$. This syndrome has weight $\leq m_T w$.
3. The signer appends k zeros in front of $\boldsymbol{s}'$: $\boldsymbol{e} = \boldsymbol{0}_k || \boldsymbol{s}'$ where $\boldsymbol{0}_k = \underbrace{00 \cdots 0}_{k}$.
4. The signer selects m_G rows of $\boldsymbol{G}$ at random where m_G is some fixed and small constant and adds these rows to obtain a codeword $\boldsymbol{c}$ of $\mathscr{C}$ of weight $\leq w_c \stackrel{\text{def}}{=} m_G w_G$.
5. The signature is then equal to

$$\boldsymbol{\sigma} = (\boldsymbol{e} + \boldsymbol{c})\boldsymbol{S}^T. \tag{1}$$

Signature verification

1. The verifier checks that the weight of the signature $\boldsymbol{\sigma}$ is less than $(m_T w + w_c) m_S$. If this is not the case the signature is discarded.
2. He computes $\boldsymbol{s}^* \stackrel{\text{def}}{=} \mathcal{F}(\mathcal{H}(\boldsymbol{m}))$ and checks whether $\boldsymbol{H}'\boldsymbol{\sigma}^T = \boldsymbol{s}^{*T}$. If this is not the case he appends a counter l to $\mathcal{H}(\boldsymbol{m})$ and checks whether $\boldsymbol{H}'\boldsymbol{\sigma}^T = \mathcal{F}(\mathcal{H}(\boldsymbol{m})||l)^T$. If after $O(2^z)$ verification attempts no such equality is found, the signature is eventually discarded.

The point behind the verification process is the following chain of equalities

$$\begin{aligned}
\boldsymbol{H}'\boldsymbol{\sigma}^T &= \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}\boldsymbol{S}(\boldsymbol{e}^T + \boldsymbol{c}^T) \\
&= \boldsymbol{Q}^{-1}\boldsymbol{H}(\boldsymbol{e}^T + \boldsymbol{c}^T) \\
&= \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{e}^T \\
&= \boldsymbol{Q}^{-1}\boldsymbol{s}'^T \\
&= \boldsymbol{Q}^{-1}\boldsymbol{Q}\boldsymbol{s}^T \\
&= \boldsymbol{s}^T.
\end{aligned}$$

Note that this is a general description of the scheme. Now in order to have reasonable key sizes, quasi-cyclic LDGM codes and quasi-cyclic matrices $\boldsymbol{Q}$ and $\boldsymbol{S}$ are actually chosen in [BBC+13]. More precisely $\boldsymbol{G}$ is chosen as a $k_0 p \times n_0 p$ matrix formed by sparse and circulant blocks $\boldsymbol{C}_{i,j}$ of size p (and such that all the rows of $\boldsymbol{G}$ have weight w_G)

$$\boldsymbol{G} = \begin{pmatrix}
\boldsymbol{C}_{0,0} & \boldsymbol{C}_{0,1} & \boldsymbol{C}_{0,2} & \cdots & \boldsymbol{C}_{0,n_0-1} \\
\boldsymbol{C}_{1,0} & \boldsymbol{C}_{1,1} & \boldsymbol{C}_{1,2} & \cdots & \boldsymbol{C}_{1,n_0-1} \\
\boldsymbol{C}_{2,0} & \boldsymbol{C}_{2,1} & \boldsymbol{C}_{2,2} & \cdots & \boldsymbol{C}_{2,n_0-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\boldsymbol{C}_{k_0-1,0} & \boldsymbol{C}_{k_0-1,1} & \boldsymbol{C}_{k_0-1,2} & \cdots & \boldsymbol{C}_{k_0-1,n_0-1}
\end{pmatrix}.$$

[3] Here || stands for the concatenation of strings.

Moreover in all the parameters suggested in [BBC+13], m_T was chosen to be equal to 1, that is $\boldsymbol{T}$ is a permutation matrix. Furthermore it is assumed in [BBC+13] that $\boldsymbol{T}$ is also formed by circulant blocks of size $p \times p$ (that is $\boldsymbol{T}$ is a quasi-cyclic permutation). $\boldsymbol{R}$ is also chosen to have a block circulant form. This is obtained by choosing $\boldsymbol{R}$ as follows.

$$\boldsymbol{R} = (\boldsymbol{a}_{r_0}^T \boldsymbol{b}_{r_0}) \otimes \mathbf{1}_{p \times p}$$

where $r_0 \stackrel{\text{def}}{=} n_0 - k_0$, $\boldsymbol{a}_{r_0}$ and $\boldsymbol{b}_{r_0}$ are two binary matrices of size $z \times r_0$, $\mathbf{1}_{p \times p}$ is the all-one $p \times p$ matrix and $\otimes$ stands for the Kronecker product. This implies that $\boldsymbol{Q}$ is formed by circulant blocks of size $p \times p$. $\boldsymbol{S}$ is also chosen in this way, namely formed by circulant blocks of size $p \times p$.

3 The idea underlying the attack

3.1 Correlations between bits of the signature

The creation of a signature can be summarized as follows. It is obtained by first obtaining a binary word $\boldsymbol{s}$ of small weight from the message $\boldsymbol{m}$ that has to be signed and then computing the product

$$((\mathbf{0}_k || \boldsymbol{s}\boldsymbol{Q}^T) + \boldsymbol{c})\boldsymbol{S}^T$$

where $\boldsymbol{c}$ is a codeword of small weight $\leq w_c$ of the LDGM code chosen for this scheme. From the fact that $\boldsymbol{Q} = \boldsymbol{T} + \boldsymbol{R}$ and $\boldsymbol{R}\boldsymbol{s}^T = 0$ where $\boldsymbol{T}$ is a permutation matrix (this choice is made in all the parameters proposed in [BBC+13]), it turns out that the signature $\boldsymbol{\sigma}$ can be written as

$$\boldsymbol{\sigma} = ((\mathbf{0}_k || \boldsymbol{s}') + \boldsymbol{c})\boldsymbol{S}^T$$

where $\boldsymbol{s}'$ is a word of (small) weight w. To simplify the discussion we will make this assumption from now on, i.e. $\boldsymbol{T}$ is a permutation matrix. We wish to emphasize that this assumption is just here to simplify the discussion a little bit, and that our attack will also work in a more general setting where the weight of $\boldsymbol{s}'$ stays sufficiently small. Let us bring in the quantities

$$\boldsymbol{x} = (x_1 \dots x_n) \stackrel{\text{def}}{=} (\mathbf{0}_k || \boldsymbol{s}') + \boldsymbol{c}.$$

Here we have

$$\boldsymbol{\sigma} = \boldsymbol{x}\boldsymbol{S}^T.$$

Roughly speaking, the idea of the attack is to look for correlations between bits of $\boldsymbol{\sigma}$ by using a bunch of signatures that will allow to compute such statistics. These correlations will give a lot of useful information about $\boldsymbol{S}$ that allows to recover a column permuted version $\boldsymbol{S}_p$ of $\boldsymbol{S}$ and later on from the knowledge of $\boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}$ recover a possible matrix $\boldsymbol{Q}_p$ that allows to forge signatures.

Before we explain where these correlations come from, let us first observe that each bit σ_i of $\boldsymbol{\sigma}$ is a linear combination of a small number of bits x_j:

$$\sigma_i = \sum_{j : S_{ij} = 1} x_j \tag{2}$$

where S_{ij} denotes the entry of $\boldsymbol{S}$ at row i and column j and the bits of $\boldsymbol{x}$ are highly biased, as we have

$$\mathbf{prob}(x_i = 1) = \frac{w_c}{n} \text{ for } i \in \{1, \cdots, k\}. \tag{3}$$

$$\mathbf{prob}(x_i = 1) = \frac{w_c}{n} + \frac{w}{n-k} - 2\frac{w w_c}{n(n-k)} \approx \frac{w_c}{n} + \frac{w}{n-k} \text{ for } i \in \{k+1, \cdots, n\}. \tag{4}$$

This already allows to find for a given position i the number of x_j's for j in $\{1, \dots, k\}$ and the number of x_j's for $j \in \{k+1, \dots, n\}$ that appear in the linear combination (2) defining σ_i. For this we assume that the x_j's are independent and that their distribution is given by (3) and (4). This can be obtained by computing estimates of $\mathbf{prob}(\sigma_i = 1)$ and the piling up lemma [Mat93]

Proposition 1. *Assume that the x_j's are independent Benoulli random variables and that their distribution is given by* (3) *and* (4)*. Let*

$$l_i \stackrel{def}{=} \#\{j \in \{1,\ldots,k\} : S_{ij} = 1\}.$$
$$r_i \stackrel{def}{=} \#\{j \in \{k+1,\ldots,n\} : S_{ij} = 1\}.$$

Then

$$\mathbf{prob}(\sigma_i = 1) = \frac{1 - (1 - 2w_c/n)^{l_i + r_i}(1 - 2w/(n-k))^{r_i}}{2}$$

Computing an estimate for $\mathbf{prob}(\sigma_i = 1)$ by using a bunch of signatures allows to recover for each position the numbers l_i and r_i. This gives all of the row weights of $\boldsymbol{S}$, but we can go beyond this by taking advantage of the correlations between the bits of $\boldsymbol{\sigma}$.

If σ_i and σ_j do not share a common bit in the associated linear combination (for instance $\sigma_i = x_1 + x_{3000}$ whereas $\sigma_j = x_{1300} + x_{2780} + x_{4000}$) then we may expect that σ_i and σ_j are independent and there is no significant statistical correlation between them. On the other hand, when σ_i and σ_j share a same bit x_t in their associated linear combination (for instance $\sigma_i = x_1 + x_{3000}$ whereas $\sigma_j = x_{1300} + x_{3000} + x_{4000}$) σ_i and σ_j are clearly correlated as explained by the following proposition.

Proposition 2. *Let X_1, X_2, X_3 be independent Bernoulli variables such that $\mathbf{prob}(X_i = 1) = p_i$, $\sigma_1 \stackrel{def}{=} X_1 + X_3$ and $\sigma_2 \stackrel{def}{=} X_2 + X_3$. Then if $p_3 \notin \{0, 1\}$, $p_1, p_2 \neq \frac{1}{2}$, we have that σ_1 and σ_2 are correlated with*

$$\mathrm{Cov}(\sigma_1, \sigma_2) \stackrel{def}{=} \mathbb{E}(\sigma_1\sigma_2) - \mathbb{E}(\sigma_1)\mathbb{E}(\sigma_2) = p_3(1-p_3)(1-2p_1)(1-2p_2).$$

This proposition is proved in Section A of the appendix. By computing estimates for all $\mathrm{Cov}(\sigma_i, \sigma_j)$ we know if the associated linear combinations (2) corresponding to σ_i and σ_j share a common x_t. From this information we easily obtain $\boldsymbol{S}$ up to a column permutation as shown in Section 4.

3.2 An additional source of correlations

This method works as long as the low codewords of the LDGM code do not introduce another source of correlations that competes with the aforementioned correlations. These correlations are in essence a consequence of a rather subtle interplay between these codewords and the rows of $\boldsymbol{S}$. To explain this new source of correlations let us introduce some notation.

Notation 1. *Let (i, j) be a pair of signature positions. We denote by $n(i, j)$ the number of rows of $\boldsymbol{G}' = \boldsymbol{G}\boldsymbol{S}^T$ whose support contains both i and j. We will also use the notation n_i for the number of rows of $\boldsymbol{G}'$ whose support contains position i. Finally, we denote by $\boldsymbol{g}'_i$ the i-th row of $\boldsymbol{G}'$.*

Roughly speaking, large values of $n(i, j)$ explain the correlations between positions i and j. To understand this link, let us first observe that from (1) we know that a signature $\boldsymbol{\sigma}$ can be written as $\boldsymbol{\sigma} = (\boldsymbol{e} + \boldsymbol{c})\boldsymbol{S}^T$ where $\boldsymbol{c}$ is a sum of m_G rows of the matrix $\boldsymbol{G}$. This implies that

$$\boldsymbol{\sigma} = \sum_{s=1}^{m_G} \boldsymbol{g}'_{i_s} + \boldsymbol{e}\boldsymbol{S}^T \tag{5}$$

Here it should be noticed that the weight of the rows $\boldsymbol{g}'_i$ is rather small compared to the length n of these rows (think of about 180 for the parameters proposed for 80 bits of security in [BBC+13] whereas the length is 9600). Moreover the weight of $\boldsymbol{e}\boldsymbol{S}^T$ is approximately of the same order as the weight of the $\boldsymbol{g}'_i$'s. This means that if σ_i is equal to 1, this is generally due to the fact that one of these rows $\boldsymbol{g}'_{i_s}$ has a 1 in the i-th position. Moreover since the weight of the $\boldsymbol{g}'_i$'s is small compared to the length of these vectors, their intersection is in general very small, meaning that

if σ_i is equal to 1, this is generally due to the fact that there is exactly one of the $\boldsymbol{g}'_{i_j}$ that has an i-th coordinate equal to 1. Here (positive) correlations appear precisely when $n(i,j)$ is unusually large, that is larger than we would expect if the $\boldsymbol{g}'_i$ behaved at random. In such a case when σ_i and σ_j are both equal to 1, this is rather often due to one of those $n(i,j)$ rows $\boldsymbol{g}'_u$ that appears in both linear combinations (5) defining σ_i and σ_j (and whose i-th and j-th coordinates are both equal to 1).

To put things on a more quantitive level, we would expect that

$$\mathbb{E}(n(i,j)) = k\frac{\binom{n-2}{m_S w_G - 2}}{\binom{n}{m_S w_G}} \approx \frac{k m_S^2 w_G^2}{n^2}.$$

If $n(i,j)$ is greater than this quantity, then positive correlations appear.

Large values of $n(i,j)$ appear either because

(i) of the aforementioned phenomenon: the linear combinations (2) defining σ_i and σ_j share a common x_t. This happens when the support of the i-th row of $\boldsymbol{S}$ and the support of the j-th row of $\boldsymbol{S}$ share a common position (i.e. position t here). In such a case we denote by $n_1(i,j)$ the Hamming weight of the t-th column of $\boldsymbol{G}$.

(ii) or a certain interplay between the rows of $\boldsymbol{G}$ and the rows of $\boldsymbol{S}$ that occurs when there are rows of $\boldsymbol{G}$ whose support contains an element of the support of the i-th row of $\boldsymbol{S}$ and an element of the support of the j-th row of $\boldsymbol{S}$. We let $n_2(i,j)$ be the number of such rows.

We clearly have that the second case is more general than the first one and a row of $\boldsymbol{G}'$ is non zero in position i and position j is such that the row of the same index in $\boldsymbol{G}$ has a support that has to intersect both the support of the i-th row and the j-th row of $\boldsymbol{S}$. In other words:

$$n_2(i,j) \geq n_1(i,j) \tag{6}$$

$$n(i,j) \leq n_2(i,j). \tag{7}$$

Notice that we generally have $n(i,j) = n_2(i,j)$ and if $n_1(i,j) \neq 0$ then we generally have $n(i,j) = n_1(i,j) = n_2(i,j)$.

Correlations between σ_i and σ_j allow to detect large values of $n(i,j)$. In case (i) we obtain directly information on the rows of $\boldsymbol{S}$, however the second case does not seem to give direct information on $\boldsymbol{S}$ since it involves both $\boldsymbol{G}'$ (that we do not know) and $\boldsymbol{S}$. There is however a way to distinguish between the cases $n_1(i,j) \neq 0$ and $n_1(i,j) = 0$. This comes from the following phenomenon in the first case.

Fact 2. *Consider a column of index t of $\boldsymbol{S}$ and denote by $\{i_1, \ldots, i_s\}$ the set of rows where this column has a 1 at that position. Then all possible pairs (i_a, i_b) are correlated because σ_{i_a} and σ_{i_b} share a common x_t.*

Let us define a graph with vertex set the set of positions $\{1, \ldots, n\}$ and where two positions are linked together with an edge if they are sufficiently correlated. Of course this graph depends on the threshold we choose for deciding whether two positions are sufficiently correlated or not. Correlations of the first kind give rise to cliques associated to columns of $\boldsymbol{S}$ where the size of the clique is the weight of the column. Recall that a clique of a graph is a subset of vertices which are linked together with edges of the graph (every two distinct vertices in the clique are adjacent). The second source of correlations is unlikely to yield such cliques and this phenomenon is used in the next section to distinguish between both sources of correlations. It will be essential to recognize the first source of correlation in order to recover $\boldsymbol{S}$ up to a column permutation.

3.3 Obtaining low weight codewords of the code with parity-check matrix $\boldsymbol{H}'$

Correlations also allow to obtain codewords of the code with parity-check matrix $\boldsymbol{H}'$. Note that this code is known to an attacker since $\boldsymbol{H}'$ is public. It will be handy to introduce the following notation

Definition 1 (public code $\mathscr{C}_{\mathbf{pub}}$). *The code with parity-check matrix $\boldsymbol{H}'$ is denoted by $\mathscr{C}_{pub}$.*

We can also observe that the matrix $\boldsymbol{G}' = \boldsymbol{G}\boldsymbol{S}^T$ is a generator matrix of this code. It can be used to "perturb" signatures (by changing their Hamming weight), without changing the syndrome $\boldsymbol{H}'\boldsymbol{\sigma}^T$ of the signature. It is actually used exactly in this way in the signature scheme. Note that this is also an LDGM code since $\boldsymbol{G}'$ has rows of weight $\leq m_S w_G$. Such rows allow to add small perturbations to the signature and they are used later on in our attack.

Some of these rows can be recovered in the following fashion. Assume that we have obtained a set of valid signatures $\mathcal{S}$ and that i and j are two positions that are correlated. Consider in this case the following subset of $\mathcal{S}$:

$$\Sigma(i,j) \stackrel{\text{def}}{=} \{\text{signatures } \sigma \in \mathcal{S} : \sigma_i = \sigma_j = 1\}. \tag{8}$$

When σ_i and σ_j are significantly correlated it turns out that a non negligible fraction of elements of $\Sigma(i,j)$ are of the form $\sum_{s=1}^{m_G} \boldsymbol{g}'_{i_s} + \boldsymbol{e}\boldsymbol{S}^t$ where exactly one of those $\boldsymbol{g}'_{i_s}$ has a "1" in the i-th position and the j-th position. This means that such a $\boldsymbol{g}'_{i_s}$ is precisely one of the $n(i,j)$ rows of $\boldsymbol{G}'$ that have a 1 in the i-th and the j-th positions.

Such a phenomenon implies that if we consider the intersection of the supports of the pairs of elements σ^s and σ^t of $\Sigma(i,j)$, a fraction of order $\frac{1}{n(i,j)}$ of these intersections has an unusually large size which is precisely due to the pairs (σ^s, σ^t) that correspond to a pair of linear combinations $(\sum_{a=1}^{m_G} \boldsymbol{g}'_{i_a} + \boldsymbol{e}^s\boldsymbol{S}^t, \sum_{b=1}^{m_G} \boldsymbol{g}'_{i_b} + \boldsymbol{e}^t\boldsymbol{S}^t)$ that share a common $\boldsymbol{g}'_u$ that belongs to one of the $n(i,j)$ rows of $\boldsymbol{G}'$ that have a 1 in position i and j.

This phenomenon clearly points to an algorithm arranging signatures of $\Sigma(i,j)$ in $n(i,j)$ groups such that all the elements in a group have an unusual large intersection with each other. Each group corresponds here to one of the rows $\boldsymbol{g}'_l$ of $\boldsymbol{G}'$ that has a "1" in position i and j and the signatures in this group have an unusual large intersection precisely because they share this common $\boldsymbol{g}'_l$ in the linear combination (5) which defines them.

Roughly speaking, the idea of considering this set $\Sigma(i,j)$ is that it acts as a filter that gives signatures for which an unusual number of them has a large intersection, and this because a non negligible fraction of them uses one of the rows $\boldsymbol{g}'_l$ of $\boldsymbol{G}'$ that has a "1" in position i and j in the linear combination (5) that defines them.

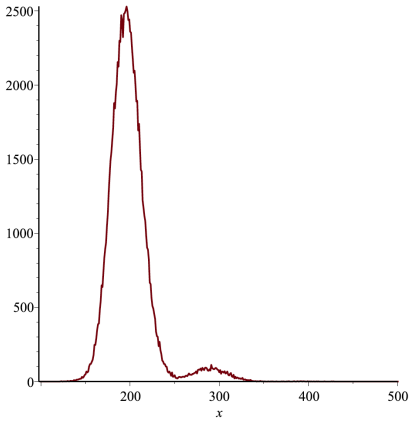
To filter inside the set $\Sigma(i,j)$ the signatures that are of this form, it suffices to compute for each signature σ in this set the number $N(\sigma)$ of signatures in $\Sigma(i,j)$ different from σ that have an unusually large intersection with σ and to keep only those signatures for which $N(\sigma)$ is large. Setting up the threshold for deciding that two signatures have a large intersection is easily achieved by plotting the histogram of those intersections as shown by Figure 1. Choosing the signatures σ of $\Sigma(i,j)$ for which $N(\sigma)$ is above this threshold yields a set that we denote by $\Sigma'(i,j)$. Then we form inside $\Sigma'(i,j)$ groups consisting of signatures which have all with each other a large intersection. This is done by considering the graph with vertices the elements of $\Sigma'(i,j)$ and putting an edge between two signatures if their intersection is sufficiently large (say greater than some threshold) and by looking for large cliques in this graph.

Once we have such groups we can recover from them some of the rows of $\boldsymbol{G}'$. Indeed, for each of those groups we can recover the common element $\boldsymbol{g}'_u$ in the linear combination (5) corresponding to these signatures. The support of $\boldsymbol{g}'_u$ is easily obtained by counting for each position i the number N_i of signatures of the group that have a 1 in this position. The support of $\boldsymbol{g}'_u$ corresponds to the positions i for which N_i is unusually large.

4 Recovering S up to a column permutation

Computing the correlations between bits of the signature reveals pairs (i,j) of rows of $\boldsymbol{S}$ that have a "1" at the same position. Consider a function Θ whose purpose is to give the threshold for deciding whether a pair of position (i,j) is sufficiently correlated or not. It takes five inputs : x a real number that gives the computed correlation of the pair and four nonnegative integers that

Fig. 1. Distribution of the weights of the intersections of every pair of signatures in $\Sigma(i,j)$. Here the threshold is set at a weight of about 250.



represent l_i, r_i, l_j and r_j respectively:

$$\begin{aligned}\Theta : \mathbb{R}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N} &\to \{0,1\}\\ (x,l_i,r_i,l_j,r_j) &\mapsto \Theta(x,l_i,r_i,l_j,r_j)\end{aligned}$$

In practice, it has been enough to suggest a relevant function for the "degree" $l_i + r_i$ of a position i, that is we chose a function $\Theta(x,l_i,r_i,l_j,r_j)$ depending only on x, l_i+r_i and l_j+r_j. We associate to such a threshold function Θ a graph $\mathcal{G}_\Theta$ defined as follows

Definition 2 (Threshold graph). *The threshold graph $\mathcal{G}_\Theta$ associated to the threshold function is the graph with*
- vertex set the set of signature positions,
- there is an edge between i and j if and only if $\Theta(\mathrm{empCov}(\sigma_i,\sigma_j),l_i,r_i,l_j,r_j) = 1$, where $\mathrm{empCov}(\sigma_i,\sigma_j)$ denotes the empirical covariance between σ_i and σ_j that is computed from the available set of signatures.

Let $\mathcal{G}_{\mathrm{sec}}$ be the graph with the same set of vertices and there is an edge between i and j if and only if the i-th row and j-th row of $\boldsymbol{S}$ have a "1" in common. Our aim is to recover $\mathcal{G}_{\mathrm{sec}}$ by using $\mathcal{G}_\Theta$. Note that cliques in $\mathcal{G}_{\mathrm{sec}}$ (that is subset of vertices of $\mathcal{G}_{\mathrm{sec}}$ that are all linked together by edges of $\mathcal{G}_{\mathrm{sec}}$) correspond to columns of $\boldsymbol{S}$, the clique correspond to all the rows of $\boldsymbol{S}$ where this column has a "1" entry. Recovering $\boldsymbol{S}$ up to a column permutation amounts to recover the cliques of $\mathcal{G}_{\mathrm{sec}}$.

This is easily achieved by considering two different threshold functions Θ_1 and Θ_2. The first one is chosen in a conservative manner. More precisely, we choose Θ_1 such that whenever $\Theta_1(\mathrm{empCov}(\sigma_i,\sigma_j),l_i,r_i,l_j,r_j) = 1$ there is an edge between i and j in $\mathcal{G}_{\mathrm{sec}}$ (i.e. this threshold is chosen in such a way, that if we declare that there is a correlation between i and j it always corresponds to two rows of $\boldsymbol{S}$ that have a "1" in common). To put it differently, $\mathcal{G}_{\Theta_1}$ is a subgraph of $\mathcal{G}_{\mathrm{sec}}$. The second threshold is chosen in a much less conservative way so that we never miss an edge of $\mathcal{G}_{\mathrm{sec}}$, i.e. when there is an edge between i and j in $\mathcal{G}_{\mathrm{sec}}$, then $\Theta_2(\mathrm{empCov}(\sigma_i,\sigma_j),l_i,r_i,l_j,r_j) = 1$. In other words, $\mathcal{G}_{\mathrm{sec}}$ is a subgraph of $\mathcal{G}_{\Theta_2}$ this time. In our experiments, we have always been able to choose Θ_1 and Θ_2 in this way.

Cliques of $\mathcal{G}_{\mathrm{sec}}$ are found by adding edges to $\mathcal{G}_{\Theta_1}$ and finding cliques in the "augmented" graph by closing triangles in $\mathcal{G}_{\Theta_1}$ whenever there was such a triangle in $\mathcal{G}_{\Theta_2}$. More precisely, we add an edge between i and k in $\mathcal{G}_{\Theta_1}$ when there was a j for which there are edges between i and j and between j and k in $\mathcal{G}_{\Theta_1}$ and $\{i,j,k\}$ forms a triangle in $\mathcal{G}_{\Theta_2}$ meaning that there are edges between i and j, between i and k and between j and k in $\mathcal{G}_{\Theta_2}$. We have been able to recover all cliques in $\mathcal{G}_{\mathrm{sec}}$ by this simple algorithm in all cases when the columns of $\boldsymbol{S}$ were of weight at least 3, meaning

that all vertices of $\mathcal{G}_{\text{sec}}$ are involved in at least one clique which contains a triangle. We ended up here with a matrix $\boldsymbol{S}_p$ that is equal to $\boldsymbol{S}$ up to a column permutation.

$$\boldsymbol{S}_p = \boldsymbol{S}\boldsymbol{\Pi}$$

where $\boldsymbol{\Pi}$ is a permutation matrix for which we can assume that it is formed by circulant blocks of size p (by reordering $\boldsymbol{S}_p$ in such a way that it is formed only by circulant blocks of size p).

5 Recovering $\boldsymbol{Q}$ up to a column permutation

The previous attack lead to find $\boldsymbol{S}$ up to a column permutation. This will lead us to recover $\boldsymbol{Q}$ up to a permutation too. We will need for this the following proposition.

Proposition 3. *Let $M_{r_0\times r_0}$ be the ring of $r_0p \times r_0p$ matrices formed by circulant blocks of size $p \times p$ and let $A_{r_0\times r_0}$ be the subset of matrices of $M_{r_0\times r_0}$ which are formed only by 0 blocks $\boldsymbol{0}_{p\times p}$ or by all-ones blocks $\boldsymbol{1}_{p\times p}$. $A_{r_0\times r_0}$ is a subring of $M_{r_0\times r_0}$ which is stable by multiplication*

$$A_{r_0\times r_0} M_{r_0\times r_0} = M_{r_0\times r_0} A_{r_0\times r_0} = A_{r_0\times r_0}.$$

The inverse of $\boldsymbol{Q}$ is of the form $\boldsymbol{T}^{-1} + \boldsymbol{A}$ where $\boldsymbol{A}$ belongs to $A_{r_0\times r_0}$.

The proof of this proposition is given in the appendix. Recall now that we have a matrix $\boldsymbol{S}_p$ which up to a column permutation is equal to $\boldsymbol{S}$, that is

$$\boldsymbol{S}_p = \boldsymbol{S}\boldsymbol{\Pi} \tag{9}$$

Recall now the following relation between the public parity-check matrix $\boldsymbol{H}'$ and the secret one $\boldsymbol{H}$:

$$\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}.$$

We also have $\boldsymbol{H} = \big(\boldsymbol{P} \mid \boldsymbol{I}\big)$. By putting all these equations together and by multiplying $\boldsymbol{H}'$ on the right by $\boldsymbol{S}_p$ we obtain

$$\begin{aligned}
\boldsymbol{H}'\boldsymbol{S}_p &= \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}\boldsymbol{S}_p \\
&= \boldsymbol{Q}^{-1}\left(\boldsymbol{P} \mid \boldsymbol{I}\right)\boldsymbol{S}^{-1}\boldsymbol{S}\boldsymbol{\Pi} \\
&= \boldsymbol{Q}^{-1}\left(\boldsymbol{P} \mid \boldsymbol{I}\right)\boldsymbol{\Pi} \\
&= \left(\boldsymbol{Q}^{-1}\boldsymbol{P} \mid \boldsymbol{Q}^{-1}\right)\boldsymbol{\Pi}
\end{aligned}$$

By using Proposition 3, we obtain

$$\boldsymbol{H}'\boldsymbol{S}_p = \big((\boldsymbol{T}^{-1} + \boldsymbol{A})\boldsymbol{P} \mid (\boldsymbol{T}^{-1} + \boldsymbol{A})\big)\,\boldsymbol{\Pi} \tag{10}$$

for some $\boldsymbol{A}$ that belongs to $A_{r_0\times r_0}$. We claim that we can find in $\boldsymbol{H}'\boldsymbol{S}'_p$ the columns that correspond to $\boldsymbol{T}^{-1} + \boldsymbol{A}$. Indeed in the matrix $\boldsymbol{A}$ the columns that belong to the same circulant block of size p are equal. Adding $\boldsymbol{T}^{-1}$ which is a permutation matrix just changes one entry per column. In other words columns belonging to the same circulant block of $\boldsymbol{T}^{-1} + \boldsymbol{A}$ are all at Hamming distance 2 from each other. Such groups of columns can easily be detected and we can find a permutation matrix $\boldsymbol{\Pi}'$ in $M_{n_0\times n_0}$ such that

$$\boldsymbol{H}'\boldsymbol{S}_p\boldsymbol{\Pi}' = \big(\boldsymbol{Q}^{-1}\boldsymbol{P}\boldsymbol{\Pi}_l \mid \boldsymbol{Q}^{-1}\boldsymbol{\Pi}_r\big) \tag{11}$$

for some permutation matrices $\boldsymbol{\Pi}_l$ and $\boldsymbol{\Pi}_r$ in $M_{k_0\times k_0}$ and $M_{r_0\times r_0}$ respectively. Then we set

$$\begin{aligned}
\boldsymbol{S}'_p &= \boldsymbol{S}_p\boldsymbol{\Pi}' \\
\boldsymbol{Q}_p &= (\boldsymbol{Q}^{-1}\boldsymbol{\Pi}_r)^{-1} = \boldsymbol{\Pi}_r^{-1}\boldsymbol{Q}
\end{aligned}$$

6 Forging new signatures

We are ready now to put all the pieces together. Forging is performed by using the pair of matrices $(\boldsymbol{Q}_p, \boldsymbol{S}'_p)$ instead of the pair $(\boldsymbol{Q}, \boldsymbol{S})$. To sign a message $\boldsymbol{m}$ we proceed as follows

1. The forger computes $\boldsymbol{s} = \mathcal{F}(\mathcal{H}(\boldsymbol{m}))$ which is an element of $\{0,1\}^{n-k}$ of weight w. He checks whether $\boldsymbol{b}\boldsymbol{s}^T = 0$. If this is the case he goes to the next step. If not, he appends a counter l to $\mathcal{H}(\boldsymbol{m})$ to obtain $\mathcal{H}(\boldsymbol{m})||l$ and computes $\mathcal{F}$ applied to $\mathcal{H}(\boldsymbol{m})||l$ until getting a syndrome $\boldsymbol{s}$ of weight w that satisfies $\boldsymbol{b}\boldsymbol{s}^T = 0$
2. He computes ${\boldsymbol{s}'}^T = \boldsymbol{Q}_p \boldsymbol{s}^T$. This syndrome has weight $\leq m_T w$
3. The forger sets $\boldsymbol{e} = \underbrace{00\cdots0}_{k}||\boldsymbol{s}'$
4. The forged signature is then computed as $\boldsymbol{\sigma} = \boldsymbol{e}{\boldsymbol{S}'_p}^T$.

It can be verified that $\boldsymbol{\sigma}$ is a valid signature since
(i) $\boldsymbol{H}'\boldsymbol{\sigma}^T = \boldsymbol{s}^T$, because

$$\begin{aligned}
\boldsymbol{H}'\boldsymbol{\sigma}^T &= \boldsymbol{H}'\boldsymbol{S}'_p\boldsymbol{e}^T \\
&= \boldsymbol{H}'\boldsymbol{S}_p\boldsymbol{\Pi}'\boldsymbol{e}^T \\
&= \left(\boldsymbol{Q}^{-1}\boldsymbol{P}\boldsymbol{\Pi}_l \mid \boldsymbol{Q}^{-1}\boldsymbol{\Pi}_r\right)\left(\boldsymbol{0}_k\ \boldsymbol{s}'\right)^T \quad \text{(follows from (11))} \\
&= \left(\boldsymbol{Q}^{-1}\boldsymbol{P}\boldsymbol{\Pi}_l \mid \boldsymbol{Q}_p^{-1}\right)\begin{pmatrix}\boldsymbol{0}_k^T \\ \boldsymbol{Q}_p\boldsymbol{s}^T\end{pmatrix} \\
&= \boldsymbol{s}^T
\end{aligned}$$

(ii) It is readily verified that the signature $\boldsymbol{\sigma}$ has Hamming weight at most $m_T m_S w$ which is smaller than $(m_T w + w_c)m_S$.

It could be argued that this weight is significantly smaller than the weight of a genuine signature which should be typically slightly less than $(m_T w + w_c)m_S$ and that this could be detected. This attack can be improved in order to achieve the "right" weight of $(m_T w + w_c)m_S$ as follows. During the recovery process of $\boldsymbol{S}$ we have found rows of $\boldsymbol{G}'$. These rows have weight of about $w_G m_S$. Since such rows are in the public code $\mathscr{C}_{\text{pub}}$ which has parity-check matrix $\boldsymbol{H}'$ we can add m_G of them to $\boldsymbol{\sigma}$ without changing the syndrome $\boldsymbol{H}'\boldsymbol{\sigma}^T$. However this adds a Hamming weight of about $m_G w_G m_S = w_c m_S$ to the signature which is precisely the weight we want to achieve.

7 Experimental Results

Running the whole attack was performed on the parameters suggested for 80 bits of security of [BBC+13] namely

n	k	p	w	w_g	w_c	z	m_T	m_s
9800	4900	50	18	20	160	2	1	9

Table 1. Parameters for 80 bits of security.

We used $100,000$ signatures to perform the attack which was implemented in `Sage` and took about one hour on a 6-core Intel® Xeon® running at 3.20 GHz. It was performed on matrices $\boldsymbol{S}$ which were either regular (constant column and row weight equal to w_S) or irregular.

8 Conclusion

We have demonstrated here that correlations between some of the bits of the signature that can be observed in the signature scheme proposed in [BBC$^+$13] can be used to recover enough of the secret information to be able to forge new signatures. Our attack was performed on the parameters devised for 80 bits of security, used $100,000$ signatures for this task and took about one hour. The real reason why this attack was possible comes from these correlations and has not to be attributed to other features of the parameters proposed in [BBC$^+$13] (for instance $\boldsymbol{T}$ was chosen as a permutation matrix, the way $\boldsymbol{S}$ is chosen is not completely specified in [BBC$^+$13] –we chose it to be either regular or irregular). Arguably, there is one place where our attack used a particular feature of the matrix $\boldsymbol{S}$, namely that its columns were at least of weight 3 –see Fact 2 where cliques in the threshold graph were used to detect correlations of type (i) (see Subsection 3.2). When $\boldsymbol{S}$ has columns of weight 1 or 2, the strategy outlined in Section 4 does not work anymore and this might require more elaborate strategies to break the scheme in such a case. It is unlikely that such a modification is able to avoid attacks using these correlations. For all these reasons, it seems to us that the scheme proposed in [BBC$^+$13] could only be used in one-time (or few-times) signature schemes.

Acknowledgment

This work was supported by the Commission of the European Communities through the Horizon 2020 program under project number 645622 PQCRYPTO. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions which were very helpful for improving the quality of the paper.

References

[BBC$^+$13] Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. Using LDGM codes and sparse syndromes to achieve digital signatures. In *Post-Quantum Cryptography 2013*, volume 7932 of *Lecture Notes in Comput. Sci.*, pages 1–15. Springer, 2013.

[BMS11] Paulo S.L.M Barreto, Rafael Misoczki, and Marcos A. Jr. Simplicio. One-time signature scheme from syndrome decoding over generic error-correcting codes. *Journal of Systems and Software*, 84(2):198–204, 2011.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 157–174, Gold Coast, Australia, 2001. Springer.

[COV07] Pierre-Louis Cayrel, Ayoub Otmani, and Damien Vergnaud. On Kabatianskii-Krouk-Smeets signatures. In *Arithmetic of Finite Fields - WAIFI 2007*, volume 4547 of *Lecture Notes in Comput. Sci.*, pages 237–251, Madrid, Spain, June 21–22 2007.

[FGO$^+$11] Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate McEliece cryptosystems. In *Proc. IEEE Inf. Theory Workshop- ITW 2011*, pages 282–286, Paraty, Brasil, October 2011.

[FGO$^+$13] Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate McEliece cryptosystems. *IEEE Trans. Inform. Theory*, 59(10):6830–6844, October 2013.

[Fin10] Matthieu Finiasz. Parallel-CFS - strengthening the CFS McEliece-based signature scheme. In *Selected Areas in Cryptography 17th International Workshop, 2010, Waterloo, Ontario, Canada, August 12-13, 2010, revised selected papers*, volume 6544 of *Lecture Notes in Comput. Sci.*, pages 159–170. Springer, 2010.

[GRSZ14] Philippe Gaborit, Olivier Ruatta, Julien Schrek, and Gilles Zémor. Ranksign: An efficient signature algorithm based on the rank metric. In *Post-Quantum Cryptography 2014*, volume 8772 of *Lecture Notes in Comput. Sci.*, pages 88–107. Springer, 2014.

[GSJB14] Danilo Gligoroski, Simona Samardjiska, Håkon Jacobsen, and Sergey Bezzateev. McEliece in the world of Escher. IACR Cryptology ePrint Archive, Report2014/360, 2014. http://eprint.iacr.org/.

[KKS97] Gregory Kabatianskii, Ernst Krouk, and Ben. J. M. Smeets. A digital signature scheme based on random error-correcting codes. In *IMA Int. Conf.*, volume 1355 of *Lecture Notes in Comput. Sci.*, pages 161–167. Springer, 1997.

[KKS05] Gregory Kabatianskii, Ernst Krouk, and Ben. J. M. Smeets. *Error Correcting Coding and Security for Data Networks: Analysis of the Superchannel Concept.* John Wiley & Sons, 2005.

[LJ12] Carl Löndahl and Thomas Johansson. A new version of McEliece PKC based on convolutional codes. In *Information and Communications Security, ICICS*, volume 7168 of *Lecture Notes in Comput. Sci.*, pages 461–470. Springer, 2012.

[LS12] Gregory Landais and Nicolas Sendrier. Implementing CFS. In *Progress in Cryptology - INDOCRYPT 2012*, volume 7668 of *Lecture Notes in Comput. Sci.*, pages 474–488. Springer, 2012.

[LT13] Grégory Landais and Jean-Pierre Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In P. Gaborit, editor, *Post-Quantum Cryptography'13*, volume 7932 of *Lecture Notes in Comput. Sci.*, pages 102–117. Springer, June 2013.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Comput. Sci.*, pages 386–397, Lofthus, Norway, May 1993. Springer.

[MS86] Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes.* North–Holland, Amsterdam, fifth edition, 1986.

[Nie86] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[OT11] Ayoub Otmani and Jean-Pierre Tillich. An efficient attack on all concrete KKS proposals. In *Post-Quantum Cryptography 2011*, volume 7071 of *Lecture Notes in Comput. Sci.*, pages 98–116, 2011.

A Proof of Proposition 2

Recall this proposition first.

Proposition. *Let X_1, X_2, X_3 be independent Bernoulli variables such that* $\mathbf{prob}(X_i = 1) = p_i$, $\sigma_1 \stackrel{def}{=} X_1 + X_3$ *and* $\sigma_2 \stackrel{def}{=} X_2 + X_3$. *Then if* $p_3 \notin \{0, 1\}$, $p_1, p_2 \neq \frac{1}{2}$, *we have that* σ_1 *and* σ_2 *are correlated with*

$$\mathrm{Cov}(\sigma_1, \sigma_2) \stackrel{def}{=} \mathbb{E}(\sigma_1\sigma_2) - \mathbb{E}(\sigma_1)\mathbb{E}(\sigma_2) = p_3(1-p_3)(1-2p_1)(1-2p_2).$$

Proof. Let us compute the probability that σ_1 and σ_2 are both equal to 1. We have

$$\begin{aligned}\mathbf{prob}(\sigma_1 = 1, \sigma_2 = 1) &= \mathbf{prob}(X_3 = 1)\mathbf{prob}(X_1 = 0)\mathbf{prob}(X_2 = 0) + \mathbf{prob}(X_3 = 0)\mathbf{prob}(X_1 = 1)\mathbf{prob}(X_2 = 1)\\ &= p_3(1-p_1)(1-p_2) + (1-p_3)p_1p_2\end{aligned}$$

On the other hand by using Proposition 1 we have

$$\begin{aligned}\mathbf{prob}(\sigma_1 = 1) &= p_1 + p_3 - 2p_1p_3\\ \mathbf{prob}(\sigma_2 = 1) &= p_2 + p_3 - 2p_2p_3\end{aligned}$$

A straighforward computation leads now to

$$\begin{aligned}\mathrm{Cov}(\sigma_1, \sigma_2) &= \mathbf{prob}(\sigma_1 = 1, \sigma_2 = 1) - \mathbf{prob}(\sigma_1 = 1)\mathbf{prob}(\sigma_2 = 1)\\ &= p_3(1-p_1)(1-p_2) + (1-p_3)p_1p_2 - (p_1 + p_3 - 2p_1p_3)(p_2 + p_3 - 2p_2p_3)\\ &= p_3\left[(1-p_1)(1-p_2) - p_1p_2 - (1-2p_1)p_2 - (1-2p_2)p_1 - (1-2p_1)(1-2p_2)p_3\right] + p_1p_2 - p_1p_2\\ &= p_3\left[1 - p_1 - p_2 + p_1p_2 - p_1p_2 - p_2 + 2p_1p_2 - p_1 + 2p_1p_2 - (1-2p_1)(1-2p_2)p_3\right]\\ &= p_3\left[1 - 2p_1 - 2p_2 + 4p_1p_2 - (1-2p_1)(1-2p_2)p_3\right]\\ &= p_3\left[(1-2p_1)(1-2p_2) - (1-2p_1)(1-2p_2)p_3\right]\\ &= p_3(1-p_3)(1-2p_1)(1-2p_2)\end{aligned}$$

B Proof of Proposition 3

Before we prove this proposition it will be very convenient to recall the following ring isomorphism Ψ between the ring of circulant binary matrices $\mathcal{M}_p$ of size $p \times p$ and $\mathbb{F}_2[X]/(1+X^p)$ which is given by

$$\Psi : \mathcal{M}_p \to \mathbb{F}_2[X]/(1+X^p)$$
$$\begin{pmatrix} a_0 & a_1 & \dots & a_{p-1} \\ a_{p-1} & a_0 & \dots & a_{p-2} \\ \dots & \dots & \ddots & \dots \\ a_1 & a_2 & \dots & a_0 \end{pmatrix} \mapsto a_0 + a_1 X + \cdots + a_{p-1} X^{p-1}$$

With this isomorphism we can view a $r_0 p \times r_0 p$ binary matrix formed by circulant blocks of size $p \times p$ as a $r_0 \times r_0$ matrix over $\mathbb{F}_2[X]/(1+X^p)$ by replacing each of these circulant blocks by its image by the isomorphism Ψ to them.

We will also use the following property of the set $C_p \stackrel{\text{def}}{=} \{0, 1+X+\cdots+X^{p-1}\}$ of $\mathbb{F}_2[X]/(X^p-1)$

Lemma 1. *C_p is an ideal of $\mathbb{F}_2[X]/(X^p-1)$.*

Proof. This is just a straighforward use of the well known theory of cyclic codes : $1+X+\cdots+X^{p-1}$ divides $1+X^p$ and C_p is nothing but the cyclic code generated by $1+X+\cdots+X^{p-1}$, see [MS86] (it is in fact a way of viewing the repetition code as a cyclic code). From this theory it follows that C_p is an ideal of $\mathbb{F}_2[X]/(X^p-1)$.

Proposition 3 can now be rephrased as

Proposition 4. *Let $M^{\psi}_{r_0 \times r_0}$ be the ring of $r_0 \times r_0$ matrices over $\mathbb{F}_2[X]/(X^p-1)$ and let $A^{\psi}_{r_0 \times r_0}$ be the ring of $r_0 \times r_0$ matrices over C_p. $A^{\psi}_{r_0 \times r_0}$ is a subring of $M^{\psi}_{r_0 \times r_0}$ which is stable by multiplication*

$$A^{\psi}_{r_0 \times r_0} M^{\psi}_{r_0 \times r_0} = M^{\psi}_{r_0 \times r_0} A^{\psi}_{r_0 \times r_0} = A^{\psi}_{r_0 \times r_0}.$$

The inverse of $\boldsymbol{Q}^{\psi}$ is of the form $(\boldsymbol{T}^{\psi})^{-1} + \boldsymbol{A}^{\psi}$ where $\boldsymbol{A}$ belongs to $A^{\psi}_{r_0 \times r_0}$, where we denote for a matrix $\boldsymbol{M}$ in $A_{r_0 \times r_0}$ by $\boldsymbol{M}^{\psi}$ the matrix where we have replaced every circulant block $\boldsymbol{M}_{ij}$ by $\psi(\boldsymbol{M}_{ij})$.

Proof. The first part follows immediately from Lemma 1. T^{ψ} is invertible and therefore

$$\begin{aligned} (\boldsymbol{Q}^{\psi})^{-1} &= (\boldsymbol{T}^{\psi} + \boldsymbol{R}^{\psi})^{-1} \\ &= (\boldsymbol{T}^{\psi})^{-1} (\boldsymbol{I} + (\boldsymbol{T}^{\psi})^{-1} \boldsymbol{R}^{\psi})^{-1} \end{aligned}$$

We use now the first part of the proposition to deduce that $\boldsymbol{A}^{\Psi} \stackrel{\text{def}}{=} (\boldsymbol{T}^{\psi})^{-1} \boldsymbol{R}^{\psi}$ belongs to $A^{\psi}_{r_0 \times r_0}$. Now it easy to prove that $(\boldsymbol{I} + \boldsymbol{A}^{\Psi})^{-1} = \boldsymbol{I} + \boldsymbol{B}^{\Psi}$ for some matrix $\boldsymbol{B}^{\Psi}$ in $A^{\Psi}_{r_0 \times r_0}$. This follows immediately from the formula

$$(\boldsymbol{I} + \boldsymbol{A}^{\Psi})^{-1} = \frac{1}{\det(\boldsymbol{I} + \boldsymbol{A}^{\psi})} \boldsymbol{C}^T$$

where $\boldsymbol{C}$ is the cofactor matrix of $\boldsymbol{I} + \boldsymbol{A}^{\Psi}$, namely the matrix where the entry c_{ij} is equal to the (i,j)-minor, that is the determinant of the $(r_0-1) \times (r_0-1)$ matrix that results from deleting row i and column j of $\boldsymbol{I} + \boldsymbol{A}^{\Psi}$. Here Lemma 1 is used to conclude that any product that contains an element of C_p yields an element in C_p. We also use the fact that any product of the form $(1+a)(1+b)$ where a and b belong to C_p is of the form $1+c$ where c belongs to C_p.